

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий  
институт

Вычислительная техника  
кафедра

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ О. В. Непомнящий  
подпись инициалы, фамилия  
« \_\_\_\_\_ » \_\_\_\_\_ 2020 г.

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

Технологии и алгоритмы обработки и хранения данных на твердотельных  
накопителях информации перспективных космических аппаратов  
тема

09.04.01 Информатика и вычислительная техника  
код и наименование направления

09.04.01.06 Микропроцессорные системы  
код и наименование магистерской программы

Научный руководитель	_____	проф, зав. каф. ВТ, канд. тех. наук.	_____	О. В. Непомнящий
	подпись, дата	должность, ученая степень		инициалы, фамилия
Выпускник	_____			М. А. Вайман
	подпись, дата			инициалы, фамилия
Рецензент	_____	ген. директор ООО Инфодор-инфо, канд. тех. наук.	_____	Э. А. Цокур
	подпись, дата	должность, ученая степень		инициалы, фамилия
Нормоконтролер	_____	проф, зав. каф. ВТ, канд. тех. наук.	_____	О. В. Непомнящий
	подпись, дата	должность, ученая степень		инициалы, фамилия

Красноярск 2020

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий  
институт

Вычислительная техника  
кафедра

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ О. В. Непомнящий  
подпись                      инициалы, фамилия  
«        »                      \_\_\_\_\_ Г.

**ЗАДАНИЕ**  
**НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**  
в форме \_\_\_\_\_ магистерской диссертации

Студенту: *Вайман Максиму Алексеевичу*

Группа: *КИ17-01-06М*      Направление (специальность): *09.04.01*  
*Информатика и вычислительная техника*

Тема выпускной квалификационной работы: *Технологии и алгоритмы обработки и хранения данных на твердотельных накопителях информации перспективных космических аппаратов.*

Утверждена приказом по университету № 4777/с от 12.04.2017.

Руководитель ВКР: *О. В. Непомнящий*, зав. каф. ВТ, канд. тех. наук., СФУ, ИКИТ, кафедра ВТ.

Исходные данные для ВКР: Выполнить анализ существующих методов сохранности целостности данных в условиях ДФКП и способов помехоустойчивого кодирования, на основе полученных результатов разработать комплекс средств противодействия ДФКП. Разработать архитектуру и функциональный состав, алгоритмы и фрагменты ПО для реконфигурируемого АПК. Провести подготовку к экспериментальным исследованиям.

Перечень разделов ВКР: Постановка задачи, моделирование, Метод комплексного, помехоустойчивого кодирования данных для твердотельных накопителей малых космических аппаратов, Разработка и тестирование лабораторного демонстратора.

Руководитель ВКР

\_\_\_\_\_

подпись

О. В. Непомнящий  
инициалы и фамилия

Задание принял в исполнение

\_\_\_\_\_

подпись

М. А. Вайман  
инициалы и фамилия

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

## **РЕФЕРАТ**

Выпускная квалификационная работа магистра по теме «Технологии и алгоритмы обработки и хранения данных на твердотельных накопителях информации перспективных космических аппаратов» содержит 45 страниц текстового документа, 43 использованных источников, 5 иллюстраций, 8 таблиц и 3 приложения.

ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ, БЛОКОВЫЕ КОДЫ, МОДЕЛЬ, СФ-БЛОК, ПЛИС, FLASH, ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС, ДЕСТАБИЛИЗИРУЮЩИЕ ФАКТОРЫ, КОСМИЧЕСКИЕ АППАРАТЫ.

Цель работы: исследование и разработка метода обеспечения целостности информации при обмене и хранении данных для твердотельных накопителей МКА, функционирующих в условиях ДФКП.

Документ содержит анализ текущего состояние предметной области, описание обоснований разработки, результаты построения математической модели, описана разработанная высокоуровневая реализация кодера/декодера, описание схем и алгоритмов функционирования .

Полученные в процессе дипломного проектирования результаты предназначены для проведения комплексного исследования и испытания в перспективных космических аппаратах.

## СОДЕРЖАНИЕ

Введение.....	3
1 Постановка задачи.....	6
1.1 Типы памяти твердотельных накопителей.....	8
1.2 Дестабилизирующие факторы космического пространства.....	10
1.3 Обзор существующих методов помехоустойчивого кодирования.....	16
1.4 Выводы.....	19
2 Метод комплексного, помехоустойчивого кодирования данных для твердотельных накопителей малых космических аппаратов.....	21
2.1 Разработка СФ-блоков для кодера/декодера.....	23
2.2 Разработка кодера и декодера.....	23
2.3 Выводы.....	27
3 Разработка и тестирование лабораторного демонстратора.....	28
3.1 Выводы.....	30
Заключение.....	31
Список сокращений.....	32
Список использованных источников.....	33
Приложение А.....	38
Приложение Б.....	42
Приложение В.....	45

## ВВЕДЕНИЕ

**Актуальность работы:** современное развитие космической отрасли диктует новые требования к объему и скорости обработки информации получаемой в процессе функционирования перспективных космических аппаратов. В данной работе под перспективными понимаются малые космические аппараты (МКА), сверх-малые КА и нано-спутники, снаряженной массой до 90 кг., функционирующие на орбитах до 1500 км (Далее по тексту под общим названием МКА).

В настоящее время для долговременного хранения бортовых данных МКА, применяется память энергонезависимого ОЗУ, функционирующего на основе ячеек магнитной или магниторезистивной памяти [1]. Данное решение, оправданно с экономической точки зрения для больших КА, однако его применение является неприемлемым для МКА в силу высокой стоимости и отсутствия широкой номенклатуры СБИС, в первую очередь отечественного производства.

Эксплуатация накопителей в составе МКА накладывает ряд ограничений связанных с масса-габаритными характеристиками, потребляемой мощности, и подверженности влиянию дестабилизирующих факторов космического пространства (ДФКП) [2].

Решение может быть найдено при использовании многоуровневой NAND-flash памяти. Такой подход будет допустим с учетом специфики применения МКА, а именно: пониженные требования к защите от воздействия ДФКП, в силу работы на низкой орбите, ограничения по энергопотреблению и масса-габаритным характеристикам.

Однако при этом следует обеспечить целостность и сохранность информации как в режимах хранения, так и при интерфейсном обмене в составе бортовых систем.

Следовательно, требуется разработка комплексного подхода к организации процесса функционирования твердотельных накопителей в составе

МКА. В данном случае необходимо применять ряд методов основанных на помехоустойчивом кодировании в режиме реального времени при информационном обмене, экранирование накопителей, дублирование или троирование основных узлов контроллера памяти, а так же алгоритмы поиска и восстановления потерянных данных в режиме хранения информации.

Исходя из выше изложенного определена основная **цель работы**: исследование и разработка метода обеспечения целостности информации при обмене и хранении данных для твердотельных накопителей МКА, функционирующих в условиях ДФКП.

Для достижения поставленной цели определены следующие основные **задачи исследования**:

1. Выполнить анализ методов обеспечения целостности информации на твердотельных накопителях (ТН), эксплуатирующихся в условиях ДФКП и разработать архитектуру специализированного контроллера для ТН.
2. Разработать алгоритмы помехоустойчивого кодирования при обмене и хранении информации на ТН. Разработать программную модель контроллера и выполнить имитационное моделирование для отработки алгоритмов и определения основных требований к аппаратному обеспечению.
3. На основе полученных результатов моделирования разработать программное обеспечение и изготовить лабораторный макет контроллера.
4. Выполнить лабораторные исследования макета и анализ полученных результатов.

**Методы исследования**: в диссертационной работе использовались основные положения теории проектирования ЭВМ и систем. При разработке ПО, использовался язык описания аппаратуры Verilog HDL и инструменты Anaconda (дистрибутив языка Python).

**Основные положения, выносимые на защиту**:

1. Разработанные алгоритмы помехоустойчивого кодирования позволяют обеспечить целостность информации при условиях и ограничениях эксплуатации, характерных для МКА;
2. Предложенный комплекс средств противодействия ДФКП в необходимой и достаточной мере способен обеспечить полное функционирование алгоритмов;
3. Разработанный и изготовленный лабораторный стенд позволяет отработать основные условия влияния ДФКП и демонстрирует работоспособность полученных технических решений.

**Научная новизна** исследования заключается в предложенном методе обеспечения целостности информации на твердотельных накопителях эксплуатирующихся в условиях ДФКП, основанном на комплексе противодействующих мер, в том числе принципа дублирования основных узлов, алгоритмах помехоустойчивого кодирования и восстановления целостности данных, реализованных на аппаратном уровне, в совокупности с технологическими приемами по экранированию позволяющих обеспечить требуемую степень защищённости от ДФКП, ёмкость накопителя и скорость обработки информации.

**Практическая ценность** результатов работы заключается в разработке комплекта сложно-функциональных блоков и встроенного программного обеспечения для контроллера NAND-Flash памяти.

**Публикации.** По результатам проведённых исследований и выполненных работ опубликована, 1 печатная работа в сборниках конференций, получено 1 регистрационное свидетельство программ для ЭВМ.

**Объём и структура диссертации.** Полный объем диссертации составляет 45 страниц, в том числе: 5 иллюстрации, 14 таблиц, 3 приложения, список литературы из 43 источника.



## 1 Постановка задачи

Одной из актуальных научно-технических задач, решаемых при проектировании современного отказоустойчивого оборудования предназначенного для освоения космического пространства, является задача передачи и хранения большого объёма информации. Данная задача является особо критичной при проектировании МКА потому, что они используются для зондирования земли, георазведки, и т. п. [3]. Это подразумевает накопление и обработку большого объёма информации, например при спутниковом ДЗЗ одна фотография, с высоким разрешением, может иметь размер в несколько гигабайт. МКА могут получать снимки с малой высоты и высоким разрешением [4]. Вывод такого аппарата не будет слишком затратным, а сам МКА не будет требовательным к условиям эксплуатации [5]. Однако МКА не может обеспечить большую полезную нагрузку, т. е. не может нести на себе широкий ассортимент аппаратуры и соответственно ни несёт на себе ёмких аккумуляторных батарей. Так же его система электропитания (солнечные панели) не может обеспечить генерацию электроэнергии большой мощности, т. е. его энергопотребление как правило не высокое, именно из-за масса-габаритных характеристик. Установка на таком аппарате мощных приём-передающих систем тоже не возможна и соответственно невозможно генерировать сигнал большой мощности и передать на высоких скоростях большой объём информации. Для таких аппаратов используются низко скоростные модемы позволяющие передать информация с минимальными энергетическими затратами [6]. Отсюда следует, что большой объём информации МКА не может достаточно быстро отправить на землю, для этого ему нужно много времени. Таким образом, актуальным является установка на МКА носителей информации большого объёма, которые долговременно могли бы хранить её.

Наибольшей популярностью не только в космосе, но и на земле, обладают именно твердотельные накопители информации. Но если на земле применение

таких накопителей особых проблем не вызывает, то в космосе это практически не возможно, поскольку такие накопители реализуются на микросхемах flash-памяти, которые в значительной мере подвержены воздействию ДФКП [7].

Тем не менее, если учесть, что МКА находится на низкой орбите, до 1500 км [8], где требования к радиационной стойкости бортовой электроники занижены (50-60 Крад по накопленной дозе) [10, 19], имеется возможность использовать Flash – как основной носитель. Кроме того, это обеспечит высокую скорость обмена информации между бортовой аппаратурой и накопителем. Таким образом для перспективного космического аппарата сформулируем следующие требования к накопителю:

1. высокая скорость записи на накопитель;
2. большой объём накопителя;
3. достаточная радиационная стойкость.

Для того чтобы обеспечить в режиме реального времени скорость записи в несколько гигабит в секунду требуется применять только аппаратные средства, т. к. ПО будет лишней прослойкой, т. е. необходим специализированный аппаратный контроллер.

Так же требуются обеспечить информационную ёмкость flash-памятью меньшей технологией изготовления микросхемы 22-60 нанометров [10]. Это позволит обеспечить уменьшение массо-габаритных характеристик и меньшее энергопотребление. Кроме того, в данном случае обеспечивается минимальное влияние эффекта единичного пробоя в результате воздействия радиационного излучения [11].

Рассмотрим проблемы обеспечения целостности информации. Для ее решения предлагается использовать комплексный подход. Во первых - экранирования накопителя и его физическое расположение в центре МКА, в составе приборов бортового комплекса управления, который в свою очередь имеет несколько слоёв экранирования. Во вторых - предлагается в процессе функционирования постоянно выполнять Refresh – циклы, то есть осуществлять

считывание и восстановление данных. В рамках данной ВКР разработан метод считывания и восстановления данных, прототип аппаратного кодера/декодера, который позволит в режиме реального времени обеспечить корректировку хранимой информации.

### **1.1 Типы памяти твердотельных накопителей**

Результаты анализа предметной области и патентного поиска позволили выделить ряд технологий энергонезависимой памяти, наиболее полно отвечающих заданию на проектирование контроллера:

- FRAM
- MRAM
- PcRAM
- ReRAM
- NVRAM

Ferroelectric random-access memory (FRAM) – Сегнетоэлектрическая память с произвольным доступом. Преимущества: низкое энергопотребление, высокая скорость записи информации, превышающее  $10^{14}$  циклов перезаписи (для устройств, рассчитанных на 3,3 В). Недостатки: низкая плотность размещения информации, ограниченная ёмкость накопителей, высокая стоимость [12].

Resistive RAM (ReRAM) – Резистивная оперативная память. Преимущества: высокая плотность, свыше  $10^6$  циклов перезаписи. Недостатки: высокая стоимость [13].

Phase-change RAM (PcRAM) – Память на основе фазового перехода. Преимущества: высокая скорость чтения информации, свыше  $10^8$  циклов перезаписи. Недостатки: высокая влияние окружающей температуры [14].

Magnetoresistive RAM (MRAM) – Магниторезистивная оперативная память. Преимущества: высокая скорость записи информации, энергопотребление. Недостатки: ограниченная ёмкость накопителей [15].

Выделены следующие типы flash-памяти по топологическому соединению и конструкции ячеек:

- NOR-flash
- NAND-flash

По конструкции одной ячейки памяти (количеству уровней заряда в ячейки):

- SLC
- MLC

Микросхемы с многоуровневыми ячейками (Multi-Level-Cell, MLC) способны хранить в одной ячейке памяти два или более битов, а микросхемы с одноуровневыми (Single-Level-Cell, SLC) – только один бит данных. Микросхемы MLC-типа, в отличие от микросхем SLC-типа, обладают намного большей емкостью, однако скорость выполнения операций чтения и записи у них ниже. Кроме того, требования к кодам, исправляющим ошибки (Error Correcting Codes, ECC), для устройств MLC-типа жестче, поскольку они в большей степени подвержены возникновению ошибок из-за плотного расположения ячеек памяти. Стоимость MLC флэш-памяти в пересчете на один бит намного меньше аналогичного параметра у SLC [16].

В таблице 1 приведены результаты сравнения рассмотренной памяти.

Таблица 1 – Основные метрики памяти для МКА

Тип памяти	Достоинства	Недостатки
FRAM	Достаточно низкое энергопотребление, высокая скорость обмена	Высокая стоимость, ограниченная информационная емкость.
MRAM	Низкое энергопотребление, высокая скорость	ограниченная информационная емкость.
РсRAM	Большое кол-во циклов	Значительно подвержена

	записи/считывания	влиянию температур
ReRAM	Высокая надёжность	Высокая стоимость
NVRAM	Низкая стоимость, не высокое энергопотребление, высокая скорость обмена	Низкая надёжность

На обосновании выше изложенного, была выбрана NAND-flash память.

## 1.2 Дестабилизирующие факторы космического пространства

Современная микроэлектронная ЭКБ, используемая в КА, должна удовлетворять целому ряду специфических требований. В процессе натурной эксплуатации КА подвергаются воздействию широкого спектра дестабилизирующих факторов космического пространства (ДФКП), каждый из которых может вызвать сбой или отказ всей системы. В общем случае необходимо принимать во внимание все регламентированные факторы.

ДФКП относятся температурное влияние окружающей среды, механическое воздействие на составные узлы МКА и радиационное воздействие. Радиационные эффекты обуславливаются влиянием ионизирующего излучения (ИИ) и тяжёлых заряженных частиц (ТЗЧ) [17,18]. ИИ в основном приводит к кратковременным эффектам, а ТЗЧ – к катастрофическим.

ИИ состоит из потока первичных заряженных частиц (электроны, протоны и тяжелые заряженные частицы), а также вторичных частиц — продуктов ядерных превращений, связанных с первичными частицами. Основные эффекты воздействия ИИ на БРЭА обусловлены ионизационными и ядерными потерями энергии первичных и вторичных частиц в чувствительных объемах элементов ИС.

Основными источниками ИИ в КП являются [19]:

- радиационный пояс Земли;
- галактическая космическая радиация;

– солнечные космические лучи.

Радиационный пояс Земли - это потоки заряженных частиц (протонов электронов), захваченных геомагнитным полем земли и образующих области повышенной радиации. Радиационный пояс Земли является основным постоянным источником радиационной опасности при полетах в околоземном пространстве.

Галактическая космическая радиация – это изотропный поток протонов и более тяжелых ядер, приходящий из удаленной области нашей Галактики или из-за ее пределов.

Солнечные космические лучи возникают в результате гигантских взрывов на поверхности Солнца, которые сопровождаются выбросом части солнечного вещества, оптическими явлениями, магнитными бурями. Состоят в основном из электронов, протонов, альфа частиц и тяжелых заряженных частиц (ТЗЧ).

Влияние ИИ КП на БРЭА сводится к двум эффектам:

- эффекту накопленной дозы (в зарубежной литературе TID — Total Ionizing Dose);
- случайным одиночным эффектам (в зарубежной литературе SEE — Single Event Effects).

Эффект накопленной дозы (ЭНД) проявляется в результате длительного воздействия ИИ, что приводит к накоплению заряда в КМОП – структуре ИС. Попадая в КМОП - структуры эти частицы ионизируют затвор и подзатворный оксид ( $\text{SiO}_2$ ). Происходит смещение пороговых напряжений и увеличение токов утечки, изменяется время нарастания и спада фронтов и т.д. Для современных микросхем актуальны радиационно-индуцированные отказы токи утечки. Под действием ионизирующего излучения в микросхеме происходит образование электронно-дырочных пар. Утечки приводят к росту тока потребления схемы (что может быть неприемлемо в космических аппаратах, где доступная мощность весьма скромна) и даже к функциональным отказам [20].

Скорость возрастания заряда уменьшается при наличии внешней защиты (оболочка КА, смежная аппаратура, стенки прибора, дополнительный алюминиевый экран и т.п.). Существуют методы расчета накопленной дозы для разных орбит в точке расположения ИС с учетом ослабления ИИ от внешней защиты, компоновки КА, конструкции приборов. Накопленная доза в кремнии ИС измеряется в радах.

Параметрический или функциональный отказ ИС наступает при достижении предельной накопленной дозы (ПНД), определяющей ее радиационную стойкость. Значение ПНД определяется при постоянно включенном питании для наиболее тяжелого нагруженного режима. Для получения данных по ПНД проводят облучение ИС на  $\gamma$  - установках до возникновения отказа при интенсивностях 110 рад/с, что на 3-4 порядка выше, чем интенсивность облучения на орбите. Для ИС и устройств, специально предназначенных для использования на КА, ПНД является одним из гарантируемых классификационных параметров. Значение ПНД дается для активного (включенного) состояния и находится в пределах 100-500 крад.

Радиационный отказ для ИС может быть обратимым, если он не вызвал дефектов структуры микросхемы или вторичных неисправностей. Накопленный под воздействием ИИ заряд с течением времени рассасывается.

Полностью восстанавливают работоспособность далеко не все ИС. В дефектных ИС за ряд остается в местах аномалий структуры. На этом основан метод радиационно-термической обработки, проводимой с целью отбора ИС с гарантированной радиационной стойкостью и повышенной надежностью. Перед использованием ИС подвергаются радиационному облучению до требуемого уровня, отказавшие ИС отбраковываются, а оставшиеся выдерживаются при температуре 100-125°C в течение 100-500 часов и еще раз отбраковываются. Накопленный заряд в радиационностойких ИС полностью рассасывается, и после термотренировки и перепроверки они могут устанавливаться в аппаратуру [21].

Второй вид отказов чаще происходит по причине одиночных случайных воздействий (ОСВ), когда тяжелые частицы попадают в ИС. В зависимости от влияния на функциональное состояние ИС выделяют следующие типы ОСВ [22]:

- кратковременные:
  1. ASET;
  2. DSET;
- остаточные:
  1. SEU;
  2. MBU;
- функциональные:
  1. SEL;
  2. SEFI;
- катастрофические:
  1. SEB;
  2. SEGR.

Кратковременные импульсы на выходах элементов ИС (SET - Single Event Transient) — изменение состояния логического выхода, вызванное ионизирующим излучением. Выделяют кратковременные импульсы на выходе цифровой ИС (Digital Single Event Transient) и кратковременные импульсы на выходе аналоговой ИС (Analog Single Event Transient). Кратковременные импульсы на выходах элементов ИС также называют “иголками”.

Одиночный сбой (SEU - Single Event Upset) является наиболее распространенным и наименее опасным последствием ионизирующего воздействия. С уменьшением проектных норм повышается чувствительность элементов к воздействию отдельной частицы. Одиночный сбой происходит при проникновении сильнодействующей частицы, типа альфа-частицы или тяжелого иона внутрь устройства. Одиночные сбои характерны для триггеров, регистров, оперативно запоминающих устройств, ПЛИС, внутренних регистровых структур больших ИС (ЦАП, АЦП, флэш память,



микроконтроллеры, микропроцессоры и т.д.) Логический элемент может воспринимать импульс тока, вызванный излучением, как сигнал данных. Если это происходит одновременно с синхроимпульса, ложный сигнал сохраняется и может быть использован другими устройствами. Несмотря на то, что пострадавший логический элемент вернется к правильному состоянию, ложный сигнал может достигнуть памяти и изменить хранящиеся в ней данные.

Множественные сбои состояния памяти (MBU - Multiple-Bit Upset) — инверсия логического состояния нескольких соседних ячеек памяти или триггера.

Единичная защелка (SEL - Single Event Latch) может быть вызвана ударом тяжелого иона в обычную КМОП интегральную схему, где существует паразитная п-р-п-р структура. SEL вызван избыточным током в базе паразитного п-р-п или р-п-р транзистора после удара тяжелого иона. Из за регенеративного цикла обратной связи, который существует между этими двумя транзисторами, защелка может произойти в пределах наносекунд и причинить разрушительное выгорание в пределах сотен микросекунд. Пороговое (сдерживающее) напряжение для единичной защелки находится обычно на уровне 1В. Таким образом, пока напряжение электропитания не снято, низкоомная структура проводимости от электропитания к земле будет сохраняться. SEL может вызвать постоянное повреждение (тяжелая ошибка) в интегральной схеме.

Одиночное функциональное прерывание (SEFI - Single Event Functional Interrupt) — аномальное поведение сложных ИС, которое возникает при инверсии логического состояния ячейки памяти или триггера управления и приводит к нарушению хода выполнения программы.

Единичное выгорание (Single Event Burnout – SEB) может произойти в мощном биполярном или мощном МОП устройстве. Мощное МОП устройство содержат паразитную биполярную транзисторную структуру. Если тяжелый ион ударяет в биполярный транзистор, сгенерированный заряд заставит ток течь в базу и поднимет потенциал перехода эмиттер–база. Если текущий ток

достаточно высок, он может сместить переход эмиттер–база в прямом направлении, и биполярный транзистор включится. После того как паразитный биполярный транзистор включен, вторичный пробой биполярного транзистора может произойти. Этот пробой может быть вызван лавинным током. В зависимости от плотности тока ток, вызванный в паразитном транзисторе тяжелым ионом, или спадает без деградации устройства, или регенеративно увеличивается до (в отсутствии ограничивающих ток элементов) разрушения устройства. Показано, что протоны также вызывают SEB.

Единичный пробой затвора (SEGR - Single Event Gate Rupture) может произойти, если тяжелый ион проходит через диэлектрик затвора. Это происходит только в высоких электрических полях типа тех, которые присутствуют в течение операции записи или очистки в энергонезависимой EEPROM ячейке или в мощном МОП транзисторе. Это вызвано объединением приложенного электрического поля и энергии, переданной частицей. Поскольку ион проходит через подзатворный окисел, то он формирует высокопроводящую плазменную дорожку между кремнием и затвором. Если энергия достаточно высока, это может вызвать локальный разогрев диэлектрика и потенциально состояние быстрого нагрева. Если это происходит, диэлектрик может локально плавиться или испаряться [23]. В таблице 2 приведены все описанные одиночные случайные воздействия.

Таблица 2 — Основные ОСВ и их воздействие на ИС

Тип эффекта	Эффект	Обозначение	Расшифровка обозначения	Влияние
Кратковременный	«Иголка»	DSET	Digital Single Event Transient	Кратковременный импульс на выходе элемента в цифровой ИС
		ASET	Analog Single Event Transient	Кратковременный импульс на выходе элемента в аналоговой ИС
Остаточный	Одиночный сбой	SEU	Single Event Upset	Изменение логического состояния ячейки памяти или триггера
	Множественные сбои	MBU	Multiple Bit Upset	Изменение логического состояния нескольких соседних ячеек памяти или регистров

Функциональный	Тиристорный эффект	SEL	Single Event Latch-up	Включение паразитной четырёхслойной рррп-структуры, приводящее к резкому увеличению тока в цепи питания
	Одиночное функциональное прерывание	SEFI	Single Event Functional Interrupt	Изменение логического состояния ячейки памяти или регистра управления, приводящее к нарушению хода выполнения программы
Катастрофический	Вторичный пробой	SEB	Single Event Burnout	Вторичный пробой рп-перехода, приводящий к его разрушению
	«Прокол диэлектрика»	SEGR	Single Event Gate Rupture	Пробой подзатворного диэлектрика вдоль трека ядерной частицы

Воздействие ИИ на БРЭА является не единственным ДФКП. Во время выведения на орбиту функционирования и эксплуатации БРЭА подвергается воздействию широкого спектра ДФКП. Согласно ГОСТ РВ 20.39.304-98 к стойкости БРЭА предъявляются определенные требования на воздействие: акустических шумов, случайных и синусоидальных вибраций, одиночных и многократных механических ударов, повышенного и пониженного давления воздуха или газа, повышенной и пониженной температуры и многих других факторов. Однако полномасштабное исследование ДФКП и их влияния на БРЭА выходит за рамки данной работы, поэтому на текущем этапе будет рассматриваться только воздействие ИИ на БРЭА.

Для того чтобы оценить влияние ИИ на БРЭА проводят специальные испытания, которые включают натурные испытания БРЭА или ее макетов, а также моделирование воздействия ИИ непосредственно на БРЭА или модели целевой тестируемой аппаратуры.

### **1.3 Обзор существующих методов помехоустойчивого кодирования**

Основа методов помехоустойчивого кодирования – это алгоритмы добавляющие к основной передаваемой в сообщении информации дополнительную избыточную информацию для контроля целостности основной

информации и возможного восстановления исходного сообщения. К таким алгоритмам относят блочные и сверточные коды.

Известны линейные циклические коды, обладающие свойством цикличности, где каждая перестановка закодированного слова по циклу является кодовым словом. К таким кодам относятся коды CRC [24], код Боуза – Чоудхури – Хоквингема [25], код Рида – Соломона [26]. Преимуществом блочных кодов является эффективность их применения при редких, сбоях с большим количестве ошибок, что как нельзя лучше подходит для нашего случая.

При реализации **сверточных кодов** информация принимается непрерывным потоком данных. Алгоритм кодирования непрерывного потока данных основан на булевой логике, когда поток подаётся на регистр сдвига, а его выходы подключены логическому элементу XOR. Декодирование производится по алгоритму Витерби. Кодирование сверточным кодом является более простой операцией по отношению к декодированию [27].

**Блочные коды** делят кодируемую информацию на блоки, преобразуя их в кодовые слова. В зависимости от алгоритма конкретного кода варьируется размер избыточной информации кодового слова, которая зависит от исходного размера информации.

В **линейных блочных кодах** информация кодируется произведением вектора исходного слова на порождающую матрицу линейного пространства. Декодирование производится таким же образом, с сравнением избыточной информации полученного сообщения.

Код Хэмминга  $(n, k)$  – является линейным блочным кодом. Где  $n$  – количество бит закодированного слова,  $k$  – количество избыточных бит. Позволяет находить и исправлять одну ошибку. Для каждого  $m$  существует  $(n=2^m-1, k=2^m-1-m)$  – код Хэмминга. Код Хэмминга можно описать с помощью приведённой реализации кодера (рисунок 1.3.1) и декодера (рисунок 1.3.2).

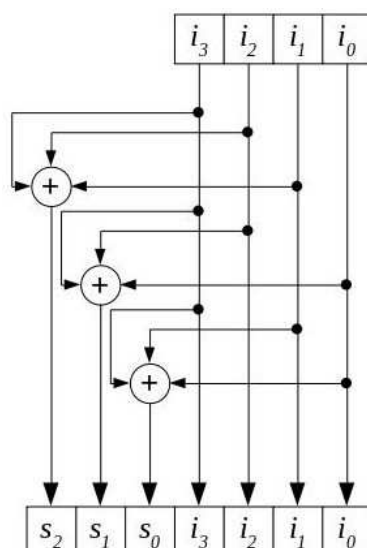


Рисунок 1.3.1 – Кодер для кода Хэмминга (7,4)

При заданных четырёх информационных разрядах  $i$ , полагаем первые четыре бита кодового слова равным этим четырём информационным битам. Дополняем тремя проверочными битами, и задавая их равенствами

$$s_1 = i_1 + i_2 + i_3$$

$$s_2 = i_2 + i_3 + i_4$$

$$s_3 = i_1 + i_2 + i_4$$

Здесь  $+$  обозначает сложение по модулю 2.

Трёхбитовая последовательность  $s$  называется синдромом. Она зависит не от истинных информационных битов, а только от конфигурации ошибок. Всего имеется восемь ( $2^3$ ) возможных синдромов: один для случая отсутствия ошибки и по одному на каждый бит закодированного слова. Простая проверка показывает, что каждая из этих ошибок имеет свой единственный синдром. Таким образом, не составляет труда сконструировать цифровую логику, которая по синдрому локализует соответствующий бит. После внесения исправления проверочные символы можно опустить. Две или более ошибки превышают возможности кодовой конструкции, и код будет ошибаться, внося не верные исправления в информационное слово. Идея этой кодовой конструкции не меняется при перестановке позиций битов в кодовых словах. Все такие варианты будут являться кодом Хэмминга(7,4).

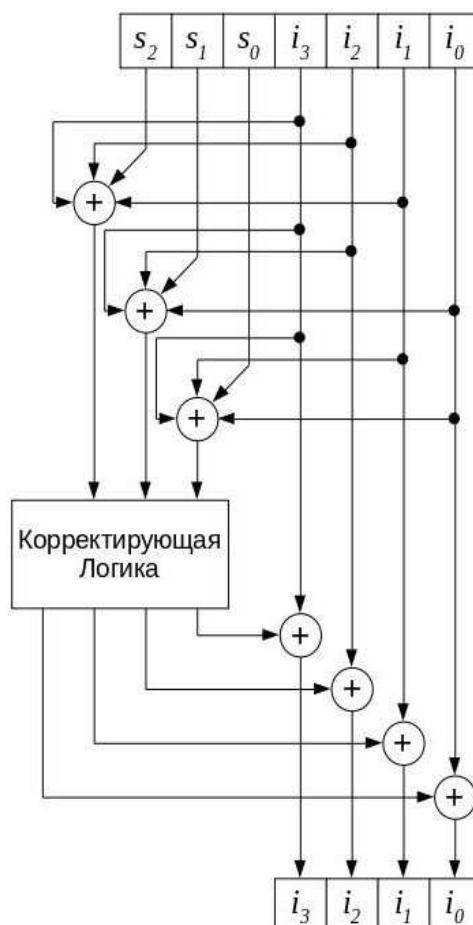


Рисунок 1.3.2 – Декодер для кода Хэмминга (7,4)

На основании анализа предметной области выделены линейные блочные коды: Код Хэмминга [28], код Рида-Маллера [29] и код Хсяо [30] для использования на аппаратном уровне в контроллере твердотельного накопителя.

#### 1.4 Выводы

Определено, что ключевыми моментами создания контроллера являются:

- использование совмещенных методов помехоустойчивого кодирования для выполнения Refresh – циклов по восстановлению информации в режиме хранения;
- обеспечение минимальных аппаратных и временных ресурсов для кодирования и декодирования данных.

Это позволило сформулировать основные требования к аппаратной реализации контроллера:

1. Накопитель должен быть реализован на базе цифровых интегральных микросхем (это обеспечит минимальные аппаратные затраты и энергопотребление).
2. Процедура кодирования/декодирования должна иметь низкую вычислительную сложность и задержку.
3. Требуется обеспечить минимальное количество избыточных символов при необходимом уровне корректирующей способности.

Таким образом, для обеспечения надёжного функционирования твердотельных накопителей в условиях ДФКП требуется разработка нового метода кодирования/декодирования информации базирующегося на комбинировании известных методов.

Так же на уровне СБИС контроллера накопителя определен перечень линейных блочных кодов: Код Хэмминга, код Рида-Маллера и код Хсяо.

## **2 Метод комплексного, помехоустойчивого кодирования данных для твердотельных накопителей малых космических аппаратов**

Предлагается реализация специализированного контроллера твердотельного накопителя реализуемого на основе ячеек NAND-Flash памяти. Контроллер предназначен для использования в бортовой аппаратуре космических аппаратов, реализуется в базисе радиационно-стойкого базового матричного кристалла (БМК) по технологическому процессу 0,18 мкм: кремний — на изоляторе [31].

Для расширения номенклатуры используемых микросхем памяти блок имеет интерфейс, совместимый с большинством современных микросхем NAND Flash-памяти. Контроллер реализован в виде комплекта сложно функциональных блоков — (IP-intellectual priority). Это позволяет использовать его для интеграции с проектами, реализуемыми по технологии «система на кристалле» (System-on-Chip, SoC). Кроме того, реализация IP из состава контроллера и тестового окружения на языке описания аппаратуры Verilog, с учетом ограничений и специфики базовых ячеек предприятия - изготовителя позволяет выпускать СБИС в базисе БМК.

На рисунке 2.1 приведен общий функциональный состав контроллера:



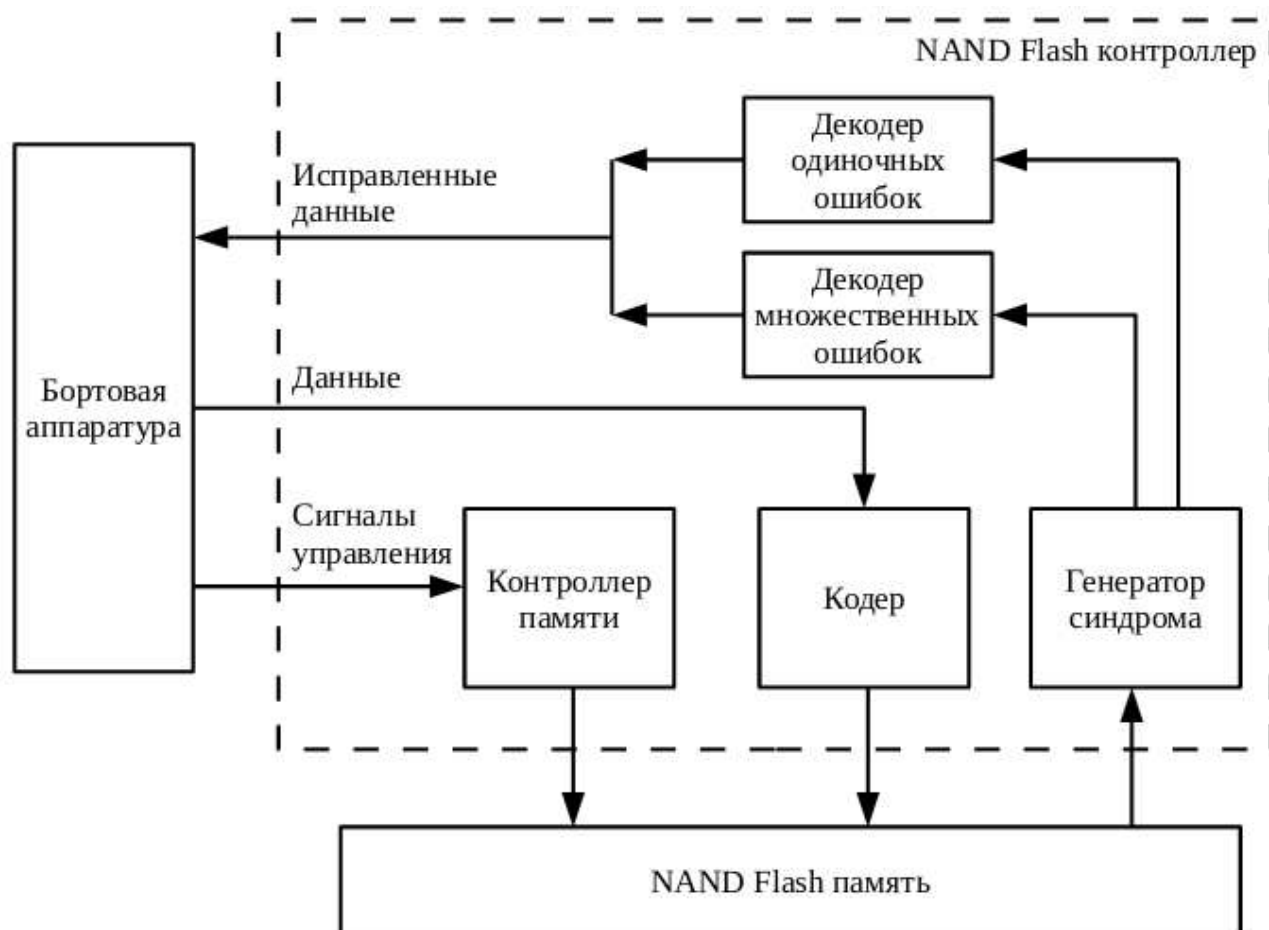


Рисунок 2.1 – Функциональный состав контроллера NAND-Flash памяти

Бортовая аппаратура генерирует поток данных для долговременного хранения. Поток данных поступает на контроллер данных. После чего контроллер памяти запускает кодер, передавая в него полученную информацию. Закодированную информацию контроллер передаёт на накопитель, структура данных накопителя предопределяется контроллером. Чтение из накопителя памяти происходит по временным меткам контроллера. Требуемая страница памяти (блок или модуль) передаётся контроллером в декодер. После этого контроллер проверяет информацию на предмет наличия ошибок, при наличии ошибки она корректируется, в случае невозможности коррекции блок помечается как дефектный и исключается из текущего сеанса. Через определённое время дефектные блоки опрашиваются, путем записи и считывания из них контрольной информации и в случае восстановления блока он включается в текущий сеанс. Контроллер так же служит для периодичной проверки структуры данных накопителя информации и работы с дефектными блоками.

В состав кодера входят три блока: Кодер Хемминга, Хсяо и Рида-Маллера. Код Хемминга даёт возможность исправить все однократные ошибки и обнаруживать все двукратные ошибки (как смежные, так и одиночные) благодаря дополнительному общему биту чётности, который вычисляется из всех информационных и кодовых бит. Содержит модуль формирования сигнала об ошибках (одинарная, двойная или ошибка в бите чётности).

Код Хсяо делает возможным исправление все однократные ошибки, обнаруживать все двукратные ошибки (как смежные, так и одиночные) и часть трёхкратных ошибок. Также содержит модуль формирования сигнала об ошибках (одинарная, двойная или тройная ошибка).

Код Рида-Маллера позволяет исправлять все однократные ошибки, двукратные и трёхкратные ошибки (как смежные, так и одиночные) и обнаруживать четырёхкратные ошибки. В данной версии СФ-блока схема формирования сигнала о количестве ошибок не реализована, однако есть сигнал, оповещающий о наличии ошибки.

## **2.1 Разработка СФ-блоков для кодера/декодера**

В соответствии с заданием на ВКР, на языке программирования Python в среде Anaconda реализованы алгоритмы кодирования/декодирования потоков данных (приложение А). Полученные результаты отработки высокоуровневых моделей позволили реализовать сложно-функциональные (СФ) блоки для кодера/декодера на языке описания аппаратуры Verilog.

## **2.2 Разработка кодера и декодера**

Опустим описание параметров кодовых конструкций и методов построения для СФ-блоков К1 и К2, так как они широко описаны в литературе [32, 33]. Однако, необходимо пояснить, что для СФ-блоков К1 и К2 была выбрана длина информационных символов 32 бита, а число проверочных 7 бит.

Наибольший интерес представляет СФ-блок КЗ, как было отмечено выше, в данном СФ-блоке реализован код Рида-Маллера второго порядка. Код Рида-Маллера  $(n, k, d_{min})$ , обозначают как РМ  $(r, m)$ , где

$n=2^m$  – длина кодового слова, при этом  $m \geq 3$  и  $r < m$ ,

$k = \sum_{i=0}^r \binom{m}{i}$  – длина информационных разрядов,

$d_{min}=2^{m-r}$  – значение минимального кодового расстояния,

$r$  – порядок кода.

Двоичные коды Рида-Маллера представляют собой семейство линейных блочных кодов, исправляющих ошибки, с декодированием, основанным на мажоритарной логике: необходимо выбрать такое подмножество проверочных уравнений, чтобы решение о значении кодового символа на определённой позиции формировалось по большинству «голосов», при этом каждый «голос» связан с одним из проверочных уравнений [34].

РМ коды образуют класс двоичных систематических кодов, имеющих различные значения:

- избыточности  $R = \frac{q}{n}$ , где  $q = n - k$  – количество избыточных (проверочных) символов;
- минимального кодового расстояния  $d_{min}$ .

Количество исправляемых ошибок определяется по следующему выражению  $t = \frac{(d_{min} - 1)}{2}$ .

В таблице 3 приведено несколько вариантов параметров для получения кода РМ  $(r, m)$  с требуемыми характеристиками.

Таблица 3 –Параметры кодов Рида-Маллера

№	$n$	$k$	$q$	$d_{min}$	$m$	$R$
1	8	4	4	4	3	0,5
2	16	11	5	4	4	0,313
3	32	16	16	8	5	0,5
4	64	42	22	8	6	0,344

Таким образом, для того чтобы код РМ  $(r, m)$  исправлял  $t \geq 2$  ошибки, были выбраны следующие параметры  $n=32$ ,  $k=16$ ,  $d_{\min}=8$ .

Рассмотрим СФ-блоки K1 и K2. Кодирующее устройство у данных СФ-блоков занимает примерно одинаковое количество комбинаторных элементов: 52 для K1; 50 для K2.

Несмотря на то, что принцип построения кода Хсяо основан на построении проверочной матрицы кода таким образом, чтобы минимизировать количество единиц в ней, из таблицы 2 видно, что СФ-блок K2 занимает бóльшую площадь чем K1, это обусловлено тем, что модуль формирования сигнала об ошибках K2 имеет более сложную структуру, так как был модифицирован, благодаря чему СФ-блок позволяет обнаруживать часть трёхкратных ошибок, при аналогичной с K1 долей избыточности и временем кодирования/декодирования. Для СФ-блоков K1 и K2 схемы соединений элементов после синтеза и размещения модуля формирования сигнала об ошибках представлены на рисунке 1, где а – модуль формирования сигнала об ошибках K1, б – модуль формирования сигнала об ошибках K2.

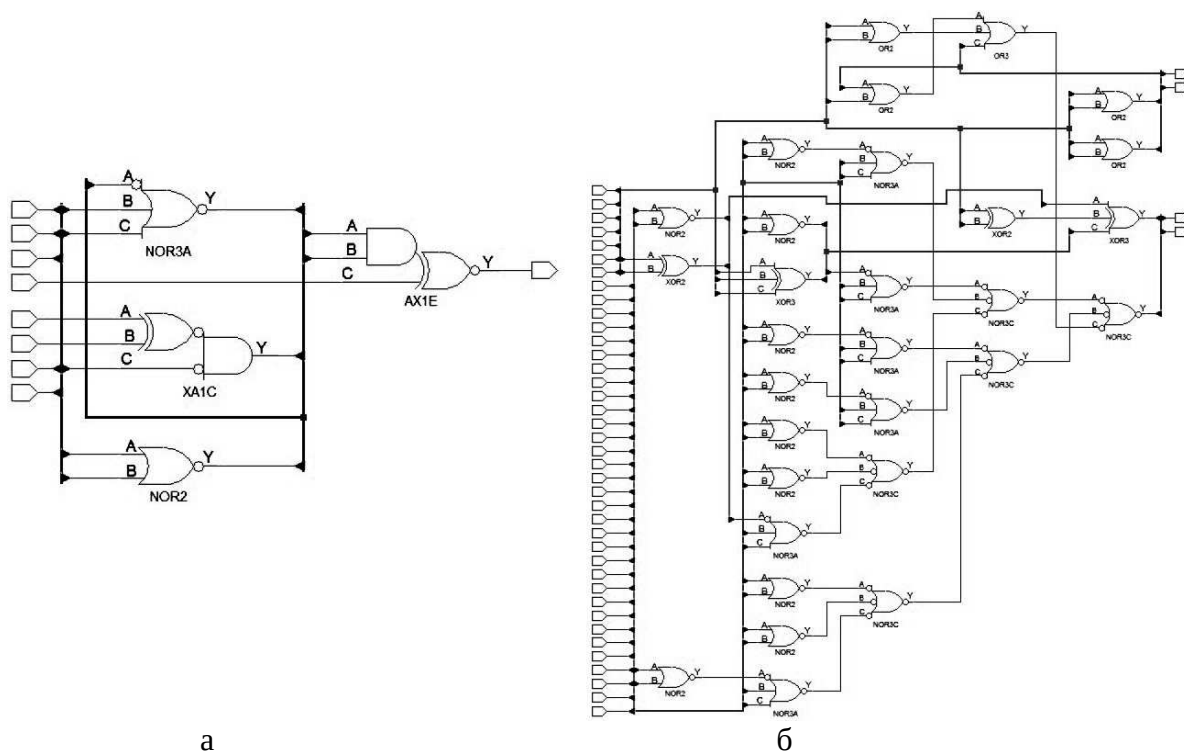


Рисунок 2.2.1 - Схема соединений элементов после синтеза

Рассмотрим СФ-блок КЗ. Несмотря на то, что порождающая матрица кода РМ (2, 5) имеет довольно большой размер ( $16_i, 32_j$ ) по сравнению с кодами Хемминга ( $6_i, 38_j$ ) и Хсяо ( $7_i, 38_j$ ), кодирующее устройство у данного СФ-блока занимает аналогичное К1 и К2 количество комбинаторных элементов: 53. Объясняется это тем, что порождающая матрица кода РМ (2, 5) имеет небольшое количество единиц, соответственно для генерации проверочных символов требуется меньше логических элементов, однако, самих проверочных символов больше чем у К1 и К2. Благодаря этому задержка от входа до выхода на операцию кодирования также сравнима с К1 и К2.

Декодирующее устройство реализовано по конвейерному принципу. Архитектура декодирующего устройства представлена на рисунке 2.2.2.

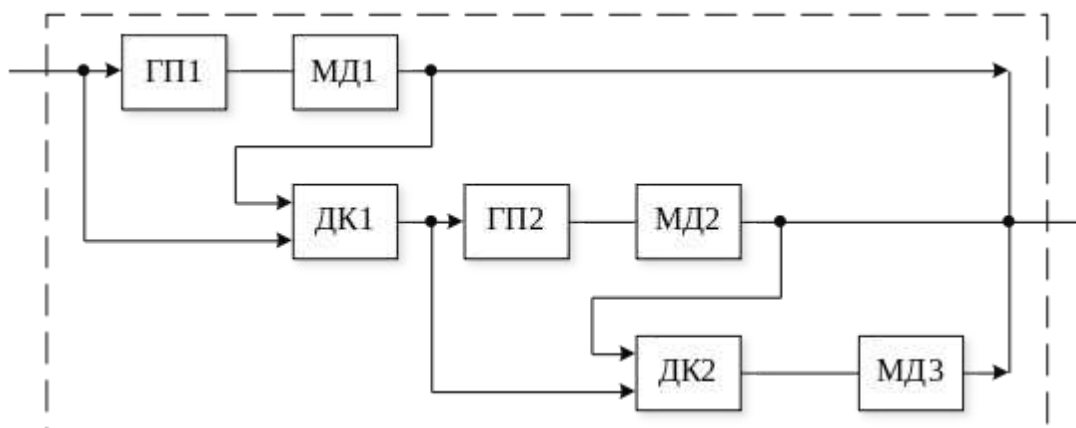


Рисунок 2.2.2 – Упрощённая структурная схема декодирующего устройства СФ-блока КЗ

Операция декодирования разделена на три этапа. На первом этапе, в генератор проверочных символов (ГП1) поступает кодовое слово  $C'$ , для первых десяти бит, которого вычисляются ортогональные проверочные суммы  $S$ . Далее мажоритарный декодер (МД1) методом «голосования» выбирает значение информационного символа на соответствующей позиции  $M_{9..0}$ . На втором этапе, в декодере (ДК1), с помощью полученного на первом этапе частично декодированного сообщения  $M_{9..0}$  и кодового слова  $C'$ , вычисляется промежуточное кодовое слово  $C''$ . Далее, в ГП2 вычисляются ортогональные

проверочные суммы для  $C''$ , которые поступают в МД2, где получаем значения следующих пяти бит информационного слова  $M_{14..10}$ . На третьем этапе, в ДК2 при помощи  $C''$  и  $M_{14..10}$  вычисляется 32-битный вектор  $C'''$ , применив к которому схожий с предыдущими этапами метод «голосования» (МД3), получим последний бит информационного сообщения  $M_{15}$ . На выходе декодера необходимо объединить части декодированного сообщения  $\{M_{15}, M_{14..10}, M_{9..0}\}$  – процесс декодирования и исправления ошибок завершён.

К недостаткам кода РМ можно отнести тот факт, что количество информационных символов  $k$  не всегда является степенью 2 (см. таблицу 3). В связи с этим, планируется модифицировать код РМ, например, методом укорочения. Данный метод не уменьшает количество проверочных символов  $q$ , а лишь позволяет подкорректировать количество информационных символов  $k$ , соответственно и длину кодового слова  $n$  под необходимую задачу.

### 2.3 Выводы

Предложен метод комплексного помехоустойчивого кодирования/декодирования данных базирующийся на принципе выбора требуемого алгоритма кодировки, в зависимости от текущего состояния накопителя.

Разработана структурная схема накопителя, программные, высокоуровневые модели СФ-блоков кодера, реализующие алгоритмы помехоустойчивого кодирования, разработаны и реализованы СФ-блоки для аппаратной реализации кодера в базисе FPGA/ASIC.

Разработано тестовое окружение, выполнен синтез и отладка СФ-блоков.

Полученные результаты позволяют перейти к разработке и изготовлению лабораторного макета твердотельного накопителя.

### 3 Разработка и тестирование лабораторного демонстратора

На данном этапе был выбран комплект для построения лабораторного стенда. Изготовленный макет имеет упрощенный состав (Рисунок 3.1).

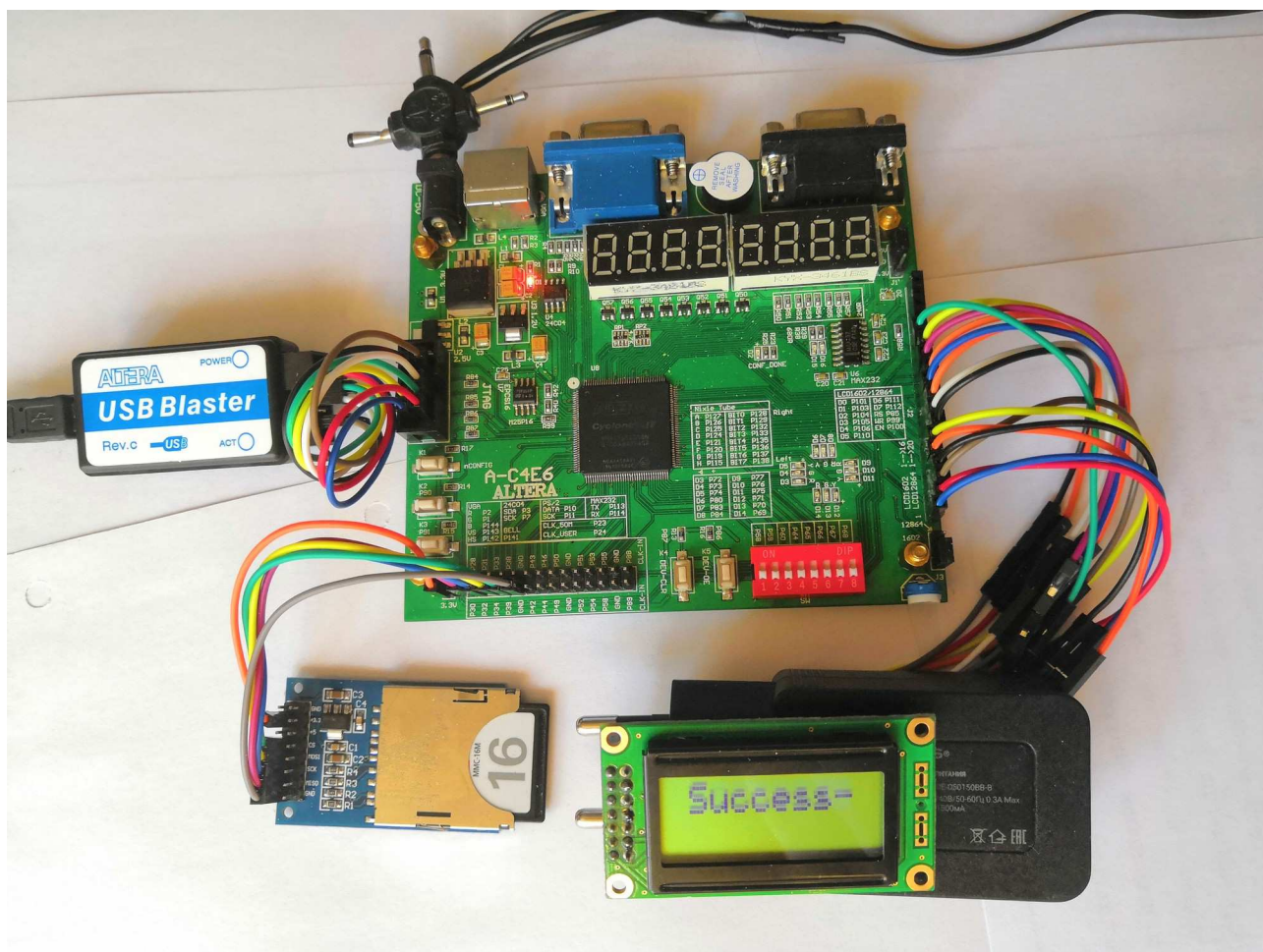


Рисунок 3.1 – Фотография макета

В состав лабораторного макета входят:

- Отладочная плата A-C4E6 FPGA Board с ПЛИС Cyclone IV EP4CE6E22C8;
- ЖКИ 8x2 Character LCD Display Module;
- Модуль SD-card;
- UART-USB.

На отладочной плате A-C4E6 установлена ПЛИС Cyclone IV EP4CE6E22C8 производства Altera Corp. (Intel), которая содержит 6272

логических ячеек, 362 логических вентилях. Так же на плате имеются следующие интерфейсы: JTAG, VGA, RS232, PS/2. Плата оборудована семисегментным индикатором на 8 символов, имеются 16 светодиодов, 8 переключателей, 5 кнопок и динамик.

Жидко-кристаллический индикатор LCD WH0802A1 состоит из 2 строк по 8 символов, управление ЖКИ осуществляется встроенным контроллером HD44780.

Модуль SD-Card работает с SD и MMC флеш-картами. Для подключения модуля используется последовательный периферийный интерфейс SPI.

Для отработки алгоритмов управления флеш-памятью и кодером разработаны и реализованы на программном уровне модули интерфейсов и управляющий модуль.

В режиме отработки алгоритмов управления накопителем и системы восстановления данных ПК выступает в роли полезной нагрузки (прибора) из состава БКУ МКА. В отличие от реального канала SpiceWare, применяемого на борту МКА, в лабораторном макете данные передаются по каналу USB. Контроллер памяти реализован на ПЛИС. При этом в качестве накопителя данных используется SD-карта, с NAND-Flash памятью. В отличие о реального прибора обмен с памятью осуществляется не в параллельном формате, а через контроллер SPI в последовательном. ЖКИ сигнализирует о выполнении операций.

Введенные упрощения не позволяют оценить быстродействие реального контроллера, но позволяют выполнить отработку основных алгоритмов функционирования.

Для оценки быстродействия требуется использовать микросхемы NAND-Flash с специальным параллельным интерфейсом, расширенным температурным диапазоном, выполненные в керамическом корпусе и предназначенные для эксплуатации в промышленных условиях. Например микросхемы от компании 3D-Plus (Франция) [35]. Кроме того, требуется



разместить низкоуровневый проект контроллера на базе радиационно-стойкой СБИС отчетственного производства, с номинальной емкостью порядка 2 млн. логических вентилей. Например БМК Алмаз-13 (5510БЦ2Т, 5510БЦ3Т) производства АО НИИМЭ и Микрон г. Зеленоград [36].

### **3.1 Выводы**

СФ-блоки кодер/декодеров разработаны на языке описания аппаратуры Verilog и протестированы на FPGA (Field-Programmable Gate Array) фирмы Intel – Cyclone IV EP4CE6E22C8.

Результаты тестирования показали, что реализованные на основе кодовых конструкций СФ-блоки, позволяют исправлять как одиночные ошибки, так и множественные.

Экспериментально определено, что не смотря на разницу в аппаратных затратах, при определённых условиях возможно добиться приблизительно одинаковых результатов по быстродействию.

## **ЗАКЛЮЧЕНИЕ**

Выполнен анализ действующих технологий и алгоритмов обеспечения целостности информации, твердотельных накопителей и условий ДФКП. В результате разработана архитектура специализированного контроллера для твердотельных накопителей.

На основе предварительного исследования были построены алгоритмы помехоустойчивого кодирования при обмене и хранении информации на ТН. С использованием математических библиотек описана модель контроллера. Для отработки алгоритмов и определения основных требований к аппаратному обеспечению произведено имитационное моделирование.

На основе полученной модели разработаны на языке описания аппаратуры IP-блоки и собран лабораторный демонстратор. Выполнены лабораторные исследования макета и анализ полученных результатов.

В продолжение данного исследования предполагается модифицировать стандартный код Рида-Маллера под необходимую длину кодового слова, что позволит сократить аппаратные затраты, уменьшить задержку на операцию декодирования и повысить корректирующую способность кода при том же количестве проверочных символов. Кроме того, необходимо рассмотреть другие перспективные кодовые конструкции, например, коды с малым количеством проверок на чётность, полярные коды и каскадные коды (гибридные).

## СПИСОК СОКРАЩЕНИЙ

- БМК — Базовый матричный кристалл
- БРЭА – Бортовая радиоэлектронная аппаратура
- ВС – Вычислительная система
- ДЗЗ – Дистанционное зондирование земли
- ДФКП – Дестабилизирующие факторы космического пространства
- ИИ – Ионизирующего излучение
- ИС – Интегральная схема
- КА – Космические аппараты
- КМОП – комплементарная логика на транзисторах металл-оксид-полупроводник
- КП – Космическое пространство
- МКА – Малые космические аппараты
- ПЗУ – Постоянное запоминающее устройство
- ПЛИС – Программируемая логическая интегральная схема
- ПНД – Предельная накопленная доза
- ПО – Программное обеспечение
- РЭА – Радио-электронная аппаратура
- СБИС – Сверхбольшая интегральная схема
- ТЗЧ – Тяжёлые заряженные частицы
- ТН – Твердотельный накопитель
- ЭВМ – Электронная вычислительная машина

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Staehle M. Current and Future Mass Memory Products // ESA Workshop on Avionics Data, Control and Software Systems (ADCSS). 2012.
2. Hsiao M.Y. A class of optimal minimum odd-weight-column SEC-DED codes // IBM J. Res. Develop. 1970. Vol. 14. P. 395–401.
3. Агафонова, И. В. Коды Рида-Маллера: примеры исправления ошибок [Электронный ресурс] / И. В. Агафонова // Семинар по дискретному гармоническому анализу и геометрическому моделированию (DHA & CAGD). – 2012. – Режим доступа: <http://dha.spb.ru/PDF/ReedMullerExamples.pdf>
4. Блейхут, Р. Теория и практика кодов, контролирующих ошибки / Р. Блейхут. – Москва : Мир, 1986. – 576 с.
5. Вернер, М. Основы кодирования : учебник / М. Вернер. – Москва : Техносфера, 2004. – 288 с.
6. Морелос-Сарагоса, Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение / Р. Морелос-Сарагоса. – Москва : Техносфера, 2006. – 320 с.
7. Gherman V., Evain S., Seymour N., et al. Generalized parity-check matrices for SEC-DED codes with fixed parity // IOLTS IEEE. 2011. P. 198–201.
8. Таперо К. И., Улимов В. Н., Членов А. М. Радиационные эффекты в кремниевых интегральных схемах космического применения. – Москва : БИНОМ. Лаборатория знаний, 2012. – 234 с.
9. Мак-Вильямс, Ф. Дж. Теория кодов, исправляющих ошибки / Ф. Дж. Мак-Вильямс, Н. Дж. А. Слоэн – Москва : Связь, 1979. – 744 с.
10. F. Wang, Single Event Upset: An Embedded Tutorial / Wang Fan, Vishwani D. Agrawal // 21st International Conference on VLSI Design, IEEE, 2008. 429-434.

11. Питерсон, У. Коды, исправляющие ошибки / У. Питерсон, Э. Уэлдон – Москва : Мир, 1976 – 590 с.
12. Чумаков А.И., Радиационная стойкость изделий ЭКБ: научное издание. М.: НИЯУ МИФИ, 2015. 512 с.
13. Уланова А.В., Васильев А.Л., Петров А.Г. Радиационные отказы микросхем флэш-памяти [Электронный ресурс] / ИЭПЭ НИЯУ МИФИ – ОАО «ЭНПО СПЭЛС» // X Межотраслевая конференция по радиационной стойкости, г. Саров. – 2012. – Режим доступа: [http://www.spels.ru/index.php?option=com\\_docman&task=doc\\_download&gid=841&Itemid=39](http://www.spels.ru/index.php?option=com_docman&task=doc_download&gid=841&Itemid=39)
14. Юдищев В. Радиационно стойкие интегральные схемы. Надежность в космосе и на Земле // Электроника: наука, технология, бизнес. 2007. №5. С. 72-77
15. Anwar M.T., Lala P.K., Thenappan P., Decoder Design for a New Single Error Correcting/Double Error Detecting Code, International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering, vol. 1, №4, 2007. 649-653.
16. Naseer R., Draper J., Parallel Double Error Correcting Code Design to Mitigate Multi-Bit Upsets in SRAMs, IEEE, 2008. 222-225.
17. Novac O., SZTRIK J., Vari-kakas S., Kim Che-Soong, Reliability Increasing Method Using a SEC-DED Hsiao Code to Cache Memories, Implemented with FPGA Circuits, Journal of Computer Science and Control Systems, vol. 4, №2, October 2011. 59-62.
18. Мальцев Г. Н., Кунгурцев В. В., Козинов И. А. Дистанционное зондирование земли на основе кластера малых космических аппаратов многоспектрального наблюдения
19. Коц Ч. Коды с исправлением ошибок и их реализация в цифровых системах / В кн.: Методы введения избыточности для вычислительных систем / Под ред. В. С. Пугачева. – М.: Сов. Радио, 1966. С. 179-229.

- 20.Хетагуров Я.А., Руднев Ю.П. Повышение надёжности цифровых устройств методами избыточного кодирования. М.: Энергия, 1974.
- 21.Гаврилов С.В., Гуров С.И., Жукова Т.Д., Рыжова Д.И., Тельпухов Д.В. Методы повышения сбоеустойчивости комбинационных ИМС методами избыточного кодирования // Прикладная математика и информатика. № 53. М.: МАКС Пресс, 2016. С. 93-102.
- 22.Poolakkarparambil M., Mathew J. BCH code based multiple bit error correction in finite field multiplier circuits. - ISQED, 2011, pp. 1-6.
- 23.Mahesh Poolakkarparambil, Jimson Mathew and Abusaleh Jabir. Multiple Bit Error Tolerant Galois Field Architectures Over GF(2<sup>m</sup>) // Electronics, 2012, 1, pp. 3-22.
- 24.Гобчанский О. Проблемы создания бортовых вычислительных комплексов малых космических аппаратов. // Современные технологии автоматизации. 2001. №4. С. 28-34
- 25.GRLIB IP Library User's Manual // Aeroflex Gaisler. 2010. P. 78
- 26.Аряшев, С. Системы на кристалле с архитектурой КОМДИВ [Электронный ресурс] / С. Аряшев // Технологии QNX и ЗОСРВ «Нейтрино» в России. – 2012. – Режим доступа: [https://www.kpda.ru/upload/iblock/5ff/aryashev\\_19.04.16.pdf](https://www.kpda.ru/upload/iblock/5ff/aryashev_19.04.16.pdf)
- 27.Kazeminejad A. Fast, minimal decoding complexity, systematic (13,8) single-error-correcting codes for on-chip DRAM applications // Electronics Letters.2001. Vol. 37(7).P. 438–440.
- 28.Vishal, A., Frost, G., Jung, D., and Newhart, D.: ‘Results to improve the efficiency of BCH and CRC codes’.
- 29.Qiang Zhang, Jun Zhou, Xiaozhou Yu. A Kind of Low-cost Non-intrusive autonomous Fault Emulation System // Computer and Information Science. 2011. Vol. 4, No. 1. P. 90–99.
- 30.Аршинов М. Н., Садовский Л. Е. Коды и математика. - М. : Наука, 1983. – 143 с.

- 31.Петров К.А. Особенности помехоустойчивого кодирования информации в ОЗУ // Электроника, микро-и нанoeлектроника. Сб.науч.тр./ под ред. В.Я. Стенина. М.: НИЯУ МИФИ, 2010. С. 167–172.
- 32.Naseer R., Draper J., DEC ECC Design to Improve Memory Reliability in Sub-100nm Technologies, IEEE, 2008. 586-589.
- 33.Maestro J.A., Reviriego P., Argyrides C., et al. Fault Tolerant Single Error Correction Encoders // Journal of Electronic Testing April. 2011.Vol. 27(2).P. 215–218.
- 34.Viterbi A.J. Convolutional Codes and Their Performance in Communication Systems // IEEE Trans. Commun. Tech. Vol. COM-19. October, 1971. P. 751–772.
- 35.Bhargava K. Efficient Implementation of Error Correction Coding In a Communication System by Using VHDL // VSRD International J. of Electrical, Electronics and Communication Engineering, Vol. 2 (6), 2012. P. 359–365.
- 36.Full-Diversity Space-time error correcting codes with low-complexity receivers. M.S. Hassan, K. Amis // EURASIP Journal on Wireless Communications and Networking Vol. 2011, Article ID 653506.
- 37.Kumar A.A., Makur A. Improved coding-theoretic and subspace-based decoding algorithms for a wider class of dct and dst codes // IEEE Transactions on Signal Processing, February 2010. P. 695–708.
- 38.Максимов И. А. Проблемы обеспечения надежного функционирования современных космических аппаратов в условиях дестабилизирующего воздействия факторов космического пространства и факторов техногенного характера // Сибирский журнал науки и технологий. 2010. №4. URL: <https://cyberleninka.ru/article/n/problemy-obespecheniya-nadezhnogo-funktsionirovaniya-sovremennyh-kosmicheskikh-apparatov-v-usloviyah-destabiliziruyuschego> (дата обращения: 12.10.2019).

- 39.Хетагуров Я.А. Руднев Ю.П. Повышение надежности цифровых устройств методами избыточного кодирования. М.: Энергия,1974.272 с
- 40.Касами Т., Токура Н., Ивадари Е., Инагаки Я. Теория кодирования. – М.: Мир, 1978.
- 41.FLASH NAND memories are the highest density memmmories qualified for space applications. // 3D Plus a HEICO company. URL: <https://www.3d-plus.com/product.php?fam=8&prod=22> (дата обращения: 27.11.2019).
- 42.IEEE Std 1364-2001 IEEE Standard Verilog Hardware Description Language
- 43.СТО 4.2–07–2014 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. – Введ. 09.01.2014. – Красноярск : ИПК СФУ, 2014. – 60 с.



## ПРИЛОЖЕНИЕ А

### Исходный код математического моделирования

```
# Code Hsiao
# Библиотеки
import numpy
# Исходные данные
n = 32 # Размер информационного слова (блока данных)
m = 7 # Размер контрольного слова
k = 39 # Размер закодированного слова

SrcData = numpy.zeros(n, bool) # Информационное слово

for i in range(1,SrcData.size,2):
    SrcData[i]= 1
print(SrcData.astype(int))
[0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1]
# Кодер
ChkData = numpy.zeros (m, bool) # Контрольные разряды
A = SrcData
B = ChkData
B[0] = A[0] ^A[1] ^A[3] ^A[4] ^A[8] ^A[9]
^A[10]^A[11]^A[17]^A[23]^A[25]^A[27]^A[31]
B[1] = A[0] ^A[5] ^A[6] ^A[8]
^A[12]^A[13]^A[14]^A[16]^A[17]^A[18]^A[19]^A[20]^A[28]
B[2] = A[1] ^A[2] ^A[5] ^A[7] ^A[9]
^A[12]^A[15]^A[20]^A[21]^A[22]^A[23]^A[25]^A[26]^A[28]
B[3] = A[2] ^A[6]
^A[10]^A[13]^A[15]^A[16]^A[24]^A[25]^A[26]^A[27]^A[28]^A[29]^A[30]^A[31]
B[4] = A[3]
^A[11]^A[16]^A[17]^A[18]^A[19]^A[20]^A[21]^A[22]^A[23]^A[26]^A[27]^A[29]^A[30]
B[5] = A[4] ^A[7] ^A[8] ^A[9]
^A[10]^A[11]^A[12]^A[13]^A[14]^A[15]^A[18]^A[21]^A[24]^A[29]
B[6] = A[0] ^A[1] ^A[2] ^A[3] ^A[4] ^A[5] ^A[6] ^A[7]
^A[14]^A[19]^A[22]^A[24]^A[30]^A[31]

print(ChkData.astype(int))

# Закодированные данные
CodeData = numpy.hstack((SrcData, ChkData))
print (CodeData.astype(int))

# Шум
CodeDataNoise = CodeData.copy()
L = 9 # Номер искажённого разряда
CodeDataNoise[L] = ~CodeDataNoise[L]
CodeDataNoise[20] = ~CodeDataNoise[10]
print (CodeDataNoise.astype(int))

# Декодер
DeCodeData = CodeDataNoise.copy()
C = DeCodeData
C[32] = C[0] ^C[1] ^C[3] ^C[4] ^C[8] ^C[9]
^C[10]^C[11]^C[17]^C[23]^C[25]^C[27]^C[31]
C[33] = C[0] ^C[5] ^C[6] ^C[8]
^C[12]^C[13]^C[14]^C[16]^C[17]^C[18]^C[19]^C[20]^C[28]
C[34] = C[1] ^C[2] ^C[5] ^C[7] ^C[9]
^C[12]^C[15]^C[20]^C[21]^C[22]^C[23]^C[25]^C[26]^C[28]
C[35] = C[2] ^C[6]
^C[10]^C[13]^C[15]^C[16]^C[24]^C[25]^C[26]^C[27]^C[28]^C[29]^C[30]^C[31]
```

```

C[36] = C[3]
^C[11]^C[16]^C[17]^C[18]^C[19]^C[20]^C[21]^C[22]^C[23]^C[26]^C[27]^C[29]^C[30]
C[37] = C[4] ^C[7] ^C[8] ^C[9]
^C[10]^C[11]^C[12]^C[13]^C[14]^C[15]^C[18]^C[21]^C[24]^C[29]
C[38] = C[0] ^C[1] ^C[2] ^C[3] ^C[4] ^C[5] ^C[6] ^C[7]
^C[14]^C[19]^C[22]^C[24]^C[30]^C[31]

print (DeCodeData.astype(int))

# Синдром
S = numpy.zeros (m, bool)
S = C[32:39]^CodeDataNoise[32:39]
print (CodeDataNoise[32:39].astype(int))
print (C[32:39].astype(int))
print (S.astype(int))

# Проверка
ErrDet = S[0]+S[1]+S[2]+S[3]+S[4]+S[5]+S[6]
ParChk = S[0]^S[1]^S[2]^S[3]^S[4]^S[5]^S[6]
#print(ErrDet, ParChk)
if ErrDet:
    print("Есть ошибка")
    if ParChk:
        print("Одиночная, Корректируется")
    else:
        print("Двойная, Не корректируется")
else:
    print("Ошибки не обнаружены")

Есть ошибка
Двойная, Не корректируется

# Нахождение ошибки в данных
D = numpy.zeros(n, bool)
#S0 1 2 3 4 5 6
H = numpy.array([[0,0,1,1,1,1,0], #0
[0,1,0,1,1,1,0], #1
[1,1,0,0,1,1,0], #2
[0,1,1,1,0,1,0], #3
[0,1,1,1,1,0,0], #4
[1,0,0,1,1,1,0], #5
[1,0,1,0,1,1,0], #6
[1,1,0,1,1,0,0], #7
[0,0,1,1,1,0,1], #8
[0,1,0,1,1,0,1], #9
[0,1,1,0,1,0,1], #10
[0,1,1,1,0,0,1], #11
[1,0,0,1,1,0,1], #12
[1,0,1,0,1,0,1], #13
[1,0,1,1,1,0,0], #14
[1,1,0,0,1,0,1], #15
[1,0,1,0,0,1,1], #16
[0,0,1,1,0,1,1], #17
[1,0,1,1,0,0,1], #18
[1,0,1,1,0,1,0], #19
[1,0,0,1,0,1,1], #20
[1,1,0,1,0,0,1], #21
[1,1,0,1,0,1,0], #22
[0,1,0,1,0,1,1], #23
[1,1,1,0,1,0,0], #24
[0,1,0,0,1,1,1], #25
[1,1,0,0,0,1,1], #26
[0,1,1,0,0,1,1], #27
[1,0,0,0,1,1,1], #28

```

```

        [1,1,1,0,0,0,1], #29
        [1,1,1,0,0,1,0], #30
        [0,1,1,0,1,1,0], #31
    ])
E = numpy.array(S ^ H)
for i in range(0,E[:,0].size,1):
    D[i]= numpy.prod(E[i])
print (D.astype(int), "- Ошибочный разряд")

# Исправление ошибки
R = numpy.array(C[0:32]^D)
print("Результат")
if ~(SrcData == R).all():
    print("Ошибка!")
print(SrcData.astype(int), "- Исходное")
print (CodeDataNoise[0:32].astype(int), "- Полученное")
print (R.astype(int), "- Декодированное")

#Code Hamming
# Библиотеки
import numpy

# Исходные данные
n = 4 # Размер информационного слова (блока данных)
m = 3 # Размер контрольного слова
k = 7 # Размер закодированного слова

SrcData = numpy.array ([0, 1, 0, 1], bool)
print(SrcData.astype(int))

# Кодер

ChkData = numpy.zeros (m, bool) # Контрольные разряды
ChkData[0] = SrcData[0] ^ SrcData[1] ^ SrcData[3]
ChkData[1] = SrcData[0] ^ SrcData[2] ^ SrcData[3]
ChkData[2] = SrcData[1] ^ SrcData[2] ^ SrcData[3]
print(ChkData.astype(int))

# Закодированные данные
CodeData = numpy.hstack((SrcData, ChkData))
print (CodeData.astype(int))

# Шум
CodeDataNoise = CodeData.copy()
L = 0 # Номер искажённого разряда
CodeDataNoise[L] = ~CodeDataNoise[L]
print (CodeDataNoise.astype(int))

# Декодер
DeCodeData = CodeDataNoise.copy()
DeCodeData[4] = DeCodeData[0] ^ DeCodeData[1] ^ DeCodeData[3]
DeCodeData[5] = DeCodeData[0] ^ DeCodeData[2] ^ DeCodeData[3]
DeCodeData[6] = DeCodeData[1] ^ DeCodeData[2] ^ DeCodeData[3]
print (DeCodeData.astype(int))

# Проверка
if ~(CodeDataNoise==DeCodeData).all():
    print("Ошибка!")
print (CodeDataNoise.astype(int))
print (DeCodeData.astype(int))

# Обнаружение ошибки
A =CodeDataNoise

```

```

B = DeCodeData
IndexErr = numpy.where(A!=B)[0]
print(IndexErr) # Номера отличающихся разрядов
# Подумать над нумерацией разрядов

# Исправление ошибки
DeCodeDataCln = CodeDataNoise.copy()
NumIndexErr = L #sum(IndexErr)
DeCodeDataCln[NumIndexErr] = ~DeCodeDataCln[NumIndexErr]
print (DeCodeDataCln.astype(int), "- Декодированное")
print (CodeData.astype(int), "- Исходное")

```

## ПРИЛОЖЕНИЕ Б

### Исходный код СФ-блоков

```
// Module LCD
module lcd0802(clk, rs, rw, en, dat);
input clk;
output [7:0] dat;
output rs, rw, en;
//tri en;
reg e;
reg [7:0] dat;
reg rs;
reg [15:0] counter;
reg [4:0] current, next;
reg clkr;
reg [1:0] cnt;
parameter set0=4'h0;
parameter set1=4'h1;
parameter set2=4'h2;
parameter set3=4'h3;
parameter dat0=4'h4;
parameter dat1=4'h5;
parameter dat2=4'h6;
parameter dat3=4'h7;
parameter dat4=4'h8;
parameter dat5=4'h9;

parameter dat6=4'hA;
parameter dat7=4'hB;
parameter dat8=4'hC;
parameter dat9=4'hD;
parameter dat10=4'hE;
parameter dat11=5'h10;
parameter nul=4'hF;
always @(posedge clk)
begin
    counter=counter+1;
    if(counter==16'h000f)
        clkr=~clkr;
end
always @(posedge clkr)
begin
    current=next;
    case(current)
        set0: begin rs<=0; dat<=8'h31; next<=set1; end
        set1: begin rs<=0; dat<=8'h0c; next<=set2; end
        set2: begin rs<=0; dat<=8'h6; next<=set3; end
        set3: begin rs<=0; dat<=8'h1; next<=dat0; end
        dat0: begin rs<=1; dat<="S"; next<=dat1; end
        dat1: begin rs<=1; dat<="u"; next<=dat2; end
        dat2: begin rs<=1; dat<="c"; next<=dat3; end
        dat3: begin rs<=1; dat<="c"; next<=dat4; end
        dat4: begin rs<=1; dat<="e"; next<=dat5; end
        dat5: begin rs<=1; dat<="s"; next<=dat6; end
        dat6: begin rs<=1; dat<="s"; next<=dat7; end
        dat7: begin rs<=1; dat<="-"; next<=dat8; end
        dat8: begin rs<=1; dat<="-"; next<=dat9; end
        dat9: begin rs<=1; dat<="-"; next<=dat10; end
        dat10: begin rs<=1; dat<="-"; next<=dat11; end
        dat11: begin rs<=1; dat<="-"; next<=nul; end
        nul: begin rs<=0; dat<=8'h00;
```

```

        if(cnt!=2'h2)
            begin
                e<=0;next<=set0;cnt<=cnt+1;
            end
        else
            begin next<=nul; e<=1;
            end
        end
    default:    next=set0;
    endcase
end
assign en=clk|e;
assign rw=0;
endmodule

// Module 7-seg
module led_7seg (clk,rst,dataout,en);

input clk,rst;
output[7:0] dataout;
output[7:0] en;

reg[7:0] dataout;
reg[7:0] en;

reg[15:0] cnt_scan;
reg[4:0] dataout_buf;

always@(posedge clk or negedge rst)
begin
    if(!rst) begin
        cnt_scan<=0;

        end
    else begin
        cnt_scan<=cnt_scan+1;
        end
    end

always @(cnt_scan)
begin
    case(cnt_scan[15:13])
        3'b000 :
            en = 8'b1111_1110;

        3'b001 :
            en = 8'b1111_1101;

        3'b010 :
            en = 8'b1111_1011;

        3'b011 :
            en = 8'b1111_0111;

        3'b100 :
            en = 8'b1110_1111;

        3'b101 :
            en = 8'b1101_1111;

        3'b110 :
            en = 8'b1011_1111;

        3'b111 :

```

```

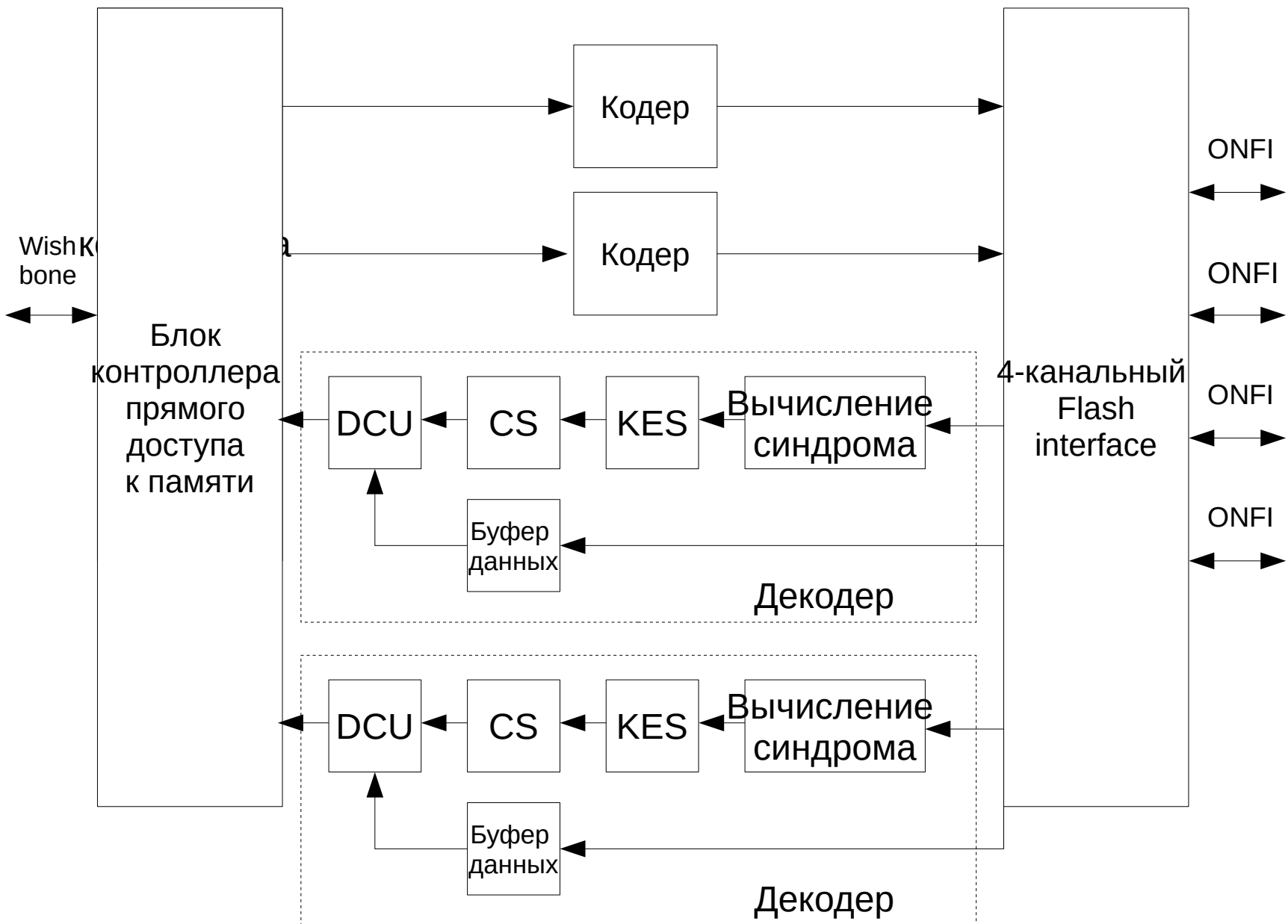
        en = 8'b0111_1111;

        default :
            en = 8'b1111_1110;
    endcase
end
always@(en)
begin
    case(en)
        8'b1111_1110:
            dataout_buf=0;
        8'b1111_1101:
            dataout_buf=1;
        8'b1111_1011:
            dataout_buf=0;
        8'b1111_0111:
            dataout_buf=1;
        8'b1110_1111:
            dataout_buf=0;
        8'b1101_1111:
            dataout_buf=1;
        8'b1011_1111:
            dataout_buf=0;
        8'b0111_1111:
            dataout_buf=1;
        default:
            dataout_buf=0;
    endcase
end

always@(dataout_buf)
begin
    case(dataout_buf)
        4'b0000:
            dataout=8'b1100_0000;
        4'b0001:
            dataout=8'b1111_1001;
        4'b0010:
            dataout=8'b1010_0100;
        4'b0011:
            dataout=8'b1011_0000;
        4'b0100:
            dataout=8'b1001_1001;
        4'b0101:
            dataout=8'b1001_0010;
        4'b0110:
            dataout=8'b1000_0010;
        4'b0111:
            dataout=8'b1111_1000;
        default:
            dataout=8'b1000_0000;
    endcase
end
endmodule

```

Структура СФ-блока контроллера NAND Flash-памяти





Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Космических и информационных технологий  
институт

Вычислительная техника  
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

подпись О. В. Непомнящий  
инициалы, фамилия  
«    » 2020 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Технологии и алгоритмы обработки и хранения данных на твердотельных  
накопителях информации перспективных космических аппаратов

тема

09.04.01 Информатика и вычислительная техника

код и наименование направления

09.04.01.06 Микропроцессорные системы

код и наименование магистерской программы

Научный руководитель

подпись, дата

проф, зав. каф. ВТ,  
канд. тех. наук.

должность, ученая степень

О. В. Непомнящий  
инициалы, фамилия

Выпускник

подпись, дата

М. А. Вайман  
инициалы, фамилия

Рецензент

подпись, дата

рецензент  
к.т.н.

должность, ученая степень

инициалы, фамилия

Нормоконтролер

подпись, дата

проф, зав. каф. ВТ,  
канд. тех. наук.

должность, ученая степень

О. В. Непомнящий  
инициалы, фамилия

Красноярск 2020